



Course Description

HTML 5 Video Game Course

Co-funded by the
Erasmus+ Programme
of the European Union



Grant agreement no.: 2014-1-DE01-KA203-000706

Project Consortium: University of Applied Sciences Magdeburg-Stendal (Germany); Aalborg University (Denmark); Lapland University of Applied Sciences (Finland); University of Lincoln (United Kingdom); University of Ljubljana (Slovenia); Potsdam University of Applied Sciences (Germany); Tampere University of Applied Sciences (Finland); University of Tampere (Finland); YMCA University of Applied Sciences (Germany)



HTML 5 Video Game Course

*Antti Perälä, Tampere University of Applied Sciences;
Björn Stockleben, Filmuniversität Babelsberg Konrad
Wolf; Sascha Reinhold, Flensburg University of Applied
Sciences; Liisa Koivisto, Lapland University of Applied
Sciences; Yrjö Koskenniemi, Lapland University of Ap-
plied Sciences*

Abstract

The video game course is a joint course between Tampere University of Applied Sciences, Filmuniversität Babelsberg Konrad Wolf, Flensburg University of Applied Sciences and Lapland University of Applied Sciences. During this online course students will learn to create HTML5-based video games. The students get a basic understanding on how to do game graphics, level design and game logic without the need to learn code syntax. The students collaborate online in order to create a working video game as the course work project.

Course Description

The students form competence-based teams in order to collaborate online for creating a HTML5 video game.

The aims of this course include:

- The students get familiar with game development concepts and tools
- The students understand the pipeline of HTML5 game development
- The students are able to bring game design ideas into working prototypes
- The students are able to arrange play testing sessions and iterate on their designs based on feedback

Course overview

The focus of the course is to HTML5 game development by example:

- 1.) Students provide a self-assessment in form of
 - A) a “trump” card and
 - B) an online-pretest.



Trump cards: https://docs.google.com/drawings/d/1CTf_WI-0-3ypJc-maulaeDZNkyhMmmugsGhDEmCLMS4U/edit

- 2.) Students form teams in a matchmaking game.
- 3.) Team choose a lightweight team collaboration platform (whatever, they just have to describe it. We might need to give some basic criteria)
- 4.) Teams choose a game genre and start developing ideas
- 5.) The students start developing their game while posting weekly video updates on their progress.
- 6.) A playtesting session is conducted in which the games are tested by an industry expert who will provide feedback to the groups.
- 7.) The games are finalized and published.

Collaboration Mode

- **Integrated Project / Virtual Team:** Students formed real virtual teams, with intensive collaboration (and intensive coordination needs) in order to collaborate on the tasks.
- **Synchronous Collaboration:** The students collaborated using live tools like chat, Slack, Concept Board, Google Docs live editing and online conferencing software (Adobe Connect, Skype).
- **Asynchronous Collaboration:** The students used asynchronous communication tools like forums, shared documents (Google Docs), shared folders (Google Drive).

Duration, Intensity

8 weeks; 3 ECTS

& ECTS

Platform

https://www.facebook.com/groups/651360411723124/?ref=group_header

Adobe Connect

Slack

Google Drive/Docs





Method

OnCreate-Method: Multidisciplinary Team Method

Other methods: Gamified Team Building: The students created trump-cards in which they identified their strengths and weaknesses.

Curriculum

The students are free to customize their learning experience by watching the amount of instructional videos they deem necessary. More independent learners might not watch the entire video library, but for those who need more support, we have a content structure like this for learning Construct 2:

1. Introductions

- 1.1 Welcome
- 1.2 Kickstarting the course: The five minute platformer
- 1.3 Is Construct 2 deep enough for developing complex games
- 1.4 Downloading and installing Construct
- 1.5 Downloading and using the manual

2. Building a basic platformer

- 2.1.1 What are we going to build
- 2.1.2 Preparations
- 2.1.3 Creating game graphics (several possible techniques)
- 2.1.4 Creating the layout
- 2.1.5 Creating the player
- 2.1.6 Testing the game
- 2.1.7 Adding game events
- 2.1.8 Creating an enemy
- 2.1.9 Creating a health system
- 2.2.1 Special movement
- 2.2.2 Creating other layouts
- 2.2.3 Advanced functionality
- 2.2.4 The saw collectible
- 2.2.5 Particle systems
- 2.2.6 Object instance health
- 2.2.7 Animating the tree falling down
- 2.2.8 Ending the game





On top of Construct 2, we also offered video lessons for the Ren'Py Visual Novel Engine, created by Sascha Reinhold. Björn Stockleben provided a video on creating game music with a tracker.

Experience Report

TAMK teacher's evaluation:

Course Overview and Introduction

It was difficult to combine the students from so many different universities because of very different schedules. We solved it by favoring asynchronous communication as much as possible. The weekly video updates made it easy for us teachers to follow the progress of the student teams.

Learning Objectives (Competencies)

The students learned the expected competencies generally well, based on the collected feedback. Most students felt competent in building a basic HTML5 video game after the course.

Assessment and Measurement

The course was assessed by the individual teachers of each university. Most universities assessed the work with a pass/fail grade based on the output of the teams. A working video game prototype and several filled surveys were required to attain the passing grade.

Instructional Materials

The instructional materials consisted of video libraries. The students had different kinds of learning flows available to them based on their individual preferences:

- Structured and gamified (watch all lessons in gamified manner)
- Free (watch lessons freely without gamification)
- Independent (watch only some lessons based on keyword queries)

The main website for the materials is www.ecation.org

Sascha Reinhold provided videos on the RenPy Visual Novel Engine and Björn Stockleben provided a video on creating game music with a tracker.

Course Activities and Learner Interaction

The learning was problem based and the students worked together in multi-disciplinary teams in order to contribute to building a working video





game prototype.

Course Technology

Facebook was the main communication platform. Besides Facebook, we used services like Youtube, Slack, Adobe Connect and Google Drive.

Learner Support

The teachers provided asynchronous support on Facebook and live support during the contact lessons.

Accessibility and Usability

The prerequisites for the course were simply a computer with an internet connection and a webcam + headset combination. No paid software was used, but only free and/or open source platforms. This made the course quite accessible to a wide range of students.

Material

Links and/or zip packages:

www.ecation.org

<https://www.facebook.com/groups/651360411723124/>

<https://www.youtube.com/playlist?list=PLy3lWYjiJZlC8ptwejrVljEfl4ZTzV-8kh>

https://www.youtube.com/watch?v=csNFaY_ywU&list=PL2ONXNjeGzh-8Kqv-oorgiNbRoYISR3mM-

<https://www.youtube.com/watch?v=yLWbp5YcNII>

